intel®

AP-380

APPLICATION NOTE

Upgrading System Designs from Bulk Erase to Boot Block Flash Memories

BRIAN DIPERT
SENIOR TECHNICAL MARKETING ENGINEER

October 1993

Order Number: 292129-001

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

Intel retains the right to make changes to these specifications at any time, without notice.

Contact your local Intel sales office or your distributor to obtain the latest specifications before placing your product order.

MDS is an ordering code only and is not used as a product name or trademark of Intel Corporation.

Intel Corporation and Intel's FASTPATH are not affiliated with Kinetics, a division of Excelan, Inc. or its FASTPATH trademark or products.

*Other brands and names are the property of their respective owners.

Additional copies of this document or other Intel literature may be obtained from:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641

or call 1-800-879-4683

Upgrading System Designs from Bulk Erase to Boot Block Flash Memories

CONTENTS

CONTENTS

1.0 INTRODUCTION

Boot Block flash memories from Intel Corporation ensure safe and simple embedded code updates for system designs. The optimized blocking of these devices locks and protects key kernel boot software while enabling easy upgrade of the majority of system code, and integrates ROM, bulk flash memory and EEPROM/NVRAM functionality in one solution. Automated algorithms simplify update routines and extend system flexibility.

This application note assists you in upgrading existing system designs based on first-generation bulk erase flash memories to Intel's advanced boot block devices. Table 1 shows a summary list of both bulk erase and boot block flash memories.

Table 1. Intel Bulk Erase and Boot Block Flash Memories

Bulk Erase	Boot Block
28F256A (x8, 256 Kbit) 32-pin DIP, PLCC	28F001BX (x8, 1 Mbit) 32-pin DIP, PLCC, TSOP
28F512 (x8, 512 Kbit) 32-pin DIP, PLCC	28F002BX (x8, 2 Mbit) 40-pin TSOP
28F010 (x8, 1 Mbit) 32-pin DIP, PLCC, TSOP	28F200BX (x8/16, 2 Mbit) 44-pin PSOP, 56-pin TSOP
28F020 (x8, 2 Mbit) 32-pin DIP, PLCC, TSOP	28F004BX (x8, 4 Mbit) 40-pin TSOP
	28F400BX (x8/16, 4 Mbit) 44-pin PSOP, 56-pin TSOP

Throughout this document, specific examples highlight conversion of a 28F010-based design to the 28F001BX. However, the techniques shown are in general applicable for any bulk erase-to-Boot Block conversion. Device datasheets (listed in Appendix L) provide comprehensive pinouts, specifications, etc. for all bulk erase and Boot Block flash memories. Issues specific to conversions other than 28F010 → 28F001BX will be spelled out where appropriate.

This application note covers the following sections:

- Pinout Compatibility
- Memory Map Compatibility
- Program and Erase Algorithm Compatibility
- DC Specification Compatibility
- AC Read Specification Compatibility
- AC Write Specification Compatibility

For additional assistance in converting your designs to Boot Block flash memories, please contact your local Intel or distribution technical sales representative.

2.0 PINOUT COMPATIBILITY

Appendices A, B and C show pinout comparisons between the 28F010 and 28F001BX flash memories. In this section, we'll discuss two key aspects of pinout compatibility;

- Added pins and functions with Boot Block devices (i.e., the RP# pin)
- Pinout adaptability between bulk erase and Boot Block memories at equivalent densities.

The RP# (Reset/Powerdown) Pin

Compared to bulk erase flash memory alternatives, Boot Block flash memories add a multi-function input called RP# (previously known as PWD#, renamed for JEDEC standardization compatibility). RP# acts as a "master on/off switch" for the flash memory, disabling a majority of internal circuitry and putting the device in an ultra-low power consumption mode when active. In this function, RP# has several distinct applications in system designs:

- It transitions the Boot Block flash memory to Deep Powerdown mode, ideal for lowest power consumption when the memory is not being accessed for an extended period. In this application, RP# is toggled by a general purpose CMOS I/O line controlled by system power management software.
- It fully protects the flash memory from unwanted command writes during system power transitions. In this application, RP# is controlled by the power supply POWERGOOD output or by discrete analog voltage monitoring circuitry.
- It resets all internal automation within the flash memory and transitions the device to the default Read Array mode. In this application, RP# is controlled by the same RESET# signal that resets the system CPU.

RP# also "unlocks" the boot block, allowing block erase and location programming, when 12V is applied to the pin.

Controlling RP#

The minimum recommended RP# control that should be implemented in Boot Block flash memory designs is shown in Figure 1. In this configuration, RP# is connected to the system power supply POWERGOOD output, or to the output of analog voltage monitoring circuitry such as the MAX70x series from Maxim Integrated Products. Additional information on flash memory write protection techniques can be found in application note AP-374, "Implementing Reliable Flash Memory Interfaces" (Order Number 292123).

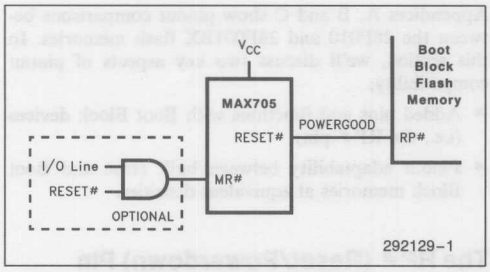


Figure 1. RP# Control

MAX70x devices (and functional equivalents from other manufacturers) also provide a MR# (manual reset) input that toggles POWERGOOD not only when the supply voltage is out-of-spec but also when MR# is active. Connecting MR# to system RESET#, therefore, allows successful CPU reboot from the flash memory even if reset occurs during program or erase operations. RP# active transitions terminate flash memory automation and return the device to Read Array mode.

If deep powerdown mode is also implemented, the MR# input to the MAX70x becomes the logical "AND" of system reset and control logic such as a system I/O line. Toggling the I/O line "low" puts the flash memory in deep powerdown mode.

In the majority of applications, kernel code stored in the boot block is programmed before the flash memory is installed on the system board. Kernel code is fully protected from alteration by providing no subsequent 12V capability on RP#. However, if future boot block update is desired, a switchable or jumpered 12V on this pin can be implemented.

Pinout Adaptability (28F010 to 28F001BX)

As Appendices A, B and C show, the package pinouts for the 28F010 and 28F001BX are very similar. The 28F010's NC (no connect) pin, reserved for address A_{17} on the 28F020, becomes RP# on the 28F001BX. Minor logic modification adds RP# control described in the previous section of this application note.

Pinout Adaptability (Other Densities)

Since the 28F256A and 28F512 are already pinout-compatible with the 28F010, upgrades from these bulk-erase flash memories to the 28F001BX are relatively straightforward. No-connects on the 28F256A and 28F512 become addresses for the higher-density 28F001BX.

Pinout conversion from the 2 Mbit 28F020 to the 28F002BX/200BX or 28F004BX/400BX is not quite as intuitive and probably requires a board re-layout. Appendices A, B and C point out that the 28F001BX uses all 32 pins in DIP, PLCC and TSOP. Higher density Boot Block flash memories, therefore, come in higher pincount packages (see Table 1), both to handle the additional addresses, and for the x16 data bus and BYTE# control of the 28F200BX and 28F400BX.

3.0 MEMORY MAP COMPATIBILITY

As their names imply, bulk erase flash memories erase all locations within the flash memory map at the same time. An example memory map for the 28F010 is shown in Figure 2.

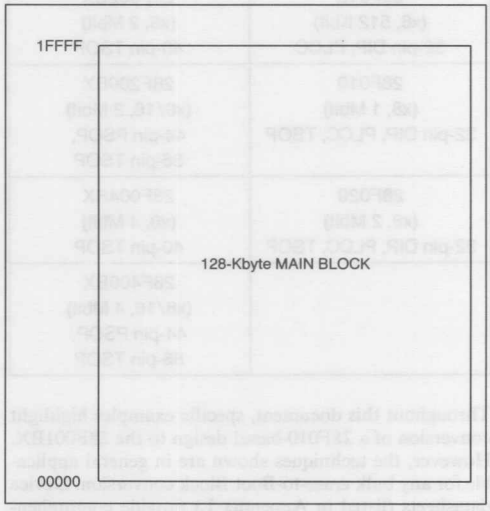


Figure 2. 28F010 Memory Map

Boot Block flash memories, on the other hand, have separately-eraseable boot, parameter and main blocks. Additionally, "top" and "bottom" versions of all Boot Block devices provide boot block locations compatible with a wide range of microprocessors and embedded processors (explained in Table 2). Figures 3 and 4 show memory maps for the 28F001BX-T ("top") and 28F001BX-B ("bottom"), respectively.

Table 2. Boot Block Flash Memory Microprocessor/Microcontroller Compatibility Chart

28F00xBX-T or 28Fx00BX-T	28F00xBX-B or 28Fx00BX-B
8086/8088	i960KA/KB
80C186/80C188 and proliferations	i960SA/SB
80286	MCS®-51 Family
Intel386™ Microprocessor Family	MCS-96 Family
Intel486™ Microprocessor Family	AMD 29K Family
Pentium™ Microprocessor	Most Motorola Microcontrollers/ Microprocessors
i860™ Microprocessor Family	
i960™CA/CF	

As first mentioned in Section 2, the boot block is intended to contain secure code which minimally will bring up the system and download code to the other blocks if required. Once initially programmed, hardware-locking the boot block from further alteration guarantees true system protection.

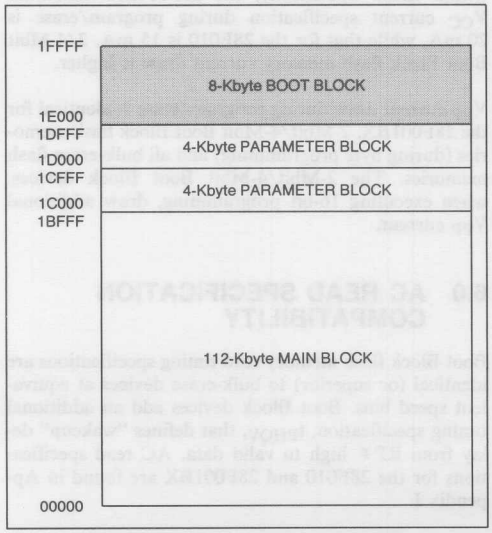


Figure 3. 28F001BX-T Memory Map

The main block(s), on the other hand, store the majority of the system code and are easily updated, since they are not hardware-lockable. Parameter blocks are intended to replace EEPROM and battery-backed SRAM for parameter storage in many designs, but these smaller blocks can also be used to store updateable system code if desired.

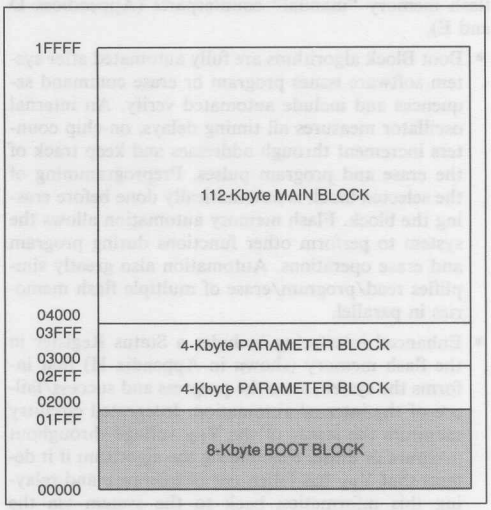


Figure 4. 28F001BX-B Memory Map

Re-segmenting your system software will allow you to take full advantage of the Boot Block architecture. The contents of "kernel" code stored in the boot block vary from system to system and application to application, but the guidelines that follow apply in most cases. The core boot code should perform some, if not all, of the following functions:

- Minimally initialize the system (configure the processor, chipset, floppy drive to allow reading in of the update code, etc.).
- Perform a "checksum" of the remainder of the flash memory data.
- If checksum verifies correctly, jump to the main portion of the boot code, found in another block of the device.
- If checksum fails (meaning that one or several of the other flash memory blocks contain invalid code/ data):
  - — Alert the user through speaker beep, message on display, LED flash, etc.
  - — Erase all other blocks of the flash memory. This means that the boot block must store the program and erase algorithms for the flash memory.
  - — Download new data from floppy disk, external connector, etc. and reprogram the other blocks.
  - — Reboot the system.

4.0 PROGRAM AND ERASE ALGORITHM COMPATIBILITY

Boot Block flash memories include second-generation automated program and erase algorithms (shown in Appendices F and G) that enhance the flash memory interface and overcome shortcomings of bulk erase flash memory "manual" counterparts (Appendices D and E).

- Boot Block algorithms are fully automated after system software issues program or erase command sequences and include automated verify. An internal oscillator measures all timing delays, on-chip counters increment through addresses and keep track of the erase and program pulses. Preprogramming of the selected block is automatically done before erasing the block. Flash memory automation allows the system to perform other functions during program and erase operations. Automation also greatly simplifies read/program/erase of multiple flash memories in parallel.
- Enhanced interfacing includes a Status Register in the flash memory (shown in Appendix H) that informs the system as to the progress and success/failure of the internal automation. Integrated circuitry monitors the status of the V_{PP} voltage throughout program or erase, terminating the algorithm if it detects that V_{PP} has fallen out of tolerance and relaying this information back to the system via the Status Register.

Although automated Boot Block flash memory program and erase **commands** are essentially backwards-compatible with those used for bulk-erase flash memories, the Boot Block flash memory automated **algorithms** themselves (although simpler) are incompatible with bulk erase manual counterparts. Compare Appendices D and F (program), and E and G (erase), to see the different algorithm steps. High-level "C" and assembly language software drivers, available for all Boot Block Flash memories, simplify algorithm development (contact your local Intel or distribution sales office).

5.0 DC SPECIFICATION COMPATIBILITY

Boot Block and bulk erase flash memories are all CMOS devices with TTL-compatible input buffers. They all require 12V V_{PP} for program and erase, and all have V_{LKO} circuitry to protect the flash memory from unintended command writes during system power transitions.

Bulk erase and Boot Block flash memories have identical CMOS current draw specifications in standby mode. Boot Block flash memories add the deep power-down mode, not found in bulk erase devices, for lowest power consumption. Versions of 2/4-Mbit Boot Block flash memories can also be operated at 3.3V V_{CC}, for lower power consumption in all operating modes (compared to 5V V_{CC} equivalent devices).

Specific DC specification differences between the 28F010 and 28F001BX are shown in Appendix I. Current draw comparison trends for bulk erase and Boot Block flash memories are described in the following sections. For particular specifications on devices other than the 28F010 and 28F001BX, consult device datasheets (listed in Appendix L).

Current Draw During Reads

The 28F001BX has identical V_{CC} and V_{PP} read current specifications to bulk erase flash memories. 2-Mbit/4-Mbit Boot Block flash memories have higher V_{CC} current specifications, reflective of their faster access times (see section 6.0) and x16 data buses. However, due to their automatic power savings feature, these devices will be generally compatible with bulk erase flash memories when similarly configured (x8) and operated at the same read frequencies.

Current Draw During Program/Erase

Boot Block flash memories will tend to consume more current through V_{CC} during program/erase compared to bulk erase devices, due to the automation running within them. Specifically, the 28F001BX maximum V_{CC} current specification during program/erase is 20 mA, while that for the 28F010 is 15 mA. 2/4-Mbit Boot Block flash memory current draw is higher.

V_{PP} current draw during program/erase is identical for the 28F001BX, 2-Mbit/4-Mbit Boot Block flash memories (during byte programming) and all bulk-erase flash memories. The 2-Mbit/4-Mbit Boot Block devices, when executing 16-bit programming, draw additional V_{PP} current.

6.0 AC READ SPECIFICATION COMPATIBILITY

Boot Block flash memory read timing specifications are identical (or superior) to bulk-erase devices at equivalent speed bins. Boot Block devices add an additional timing specification, t_{PHQV}, that defines "wakeup" delay from RP# high to valid data. AC read specifications for the 28F010 and 28F001BX are found in Appendix J.

At the time this application note was written, design "shrinks" of the 28F010, 28F020 and 28F001BX were underway to move these products to a sub-micron manufacturing process. These redesigns will allow Intel to offer faster speed bins for these devices. Intel plans to provide compatible read specifications for all existing and future versions of the 28F001BX and 28F010/28F020. Contact your local Intel or distribution sales office for more information.

7.0 AC WRITE SPECIFICATION COMPATIBILITY

In general, write specifications for matching speed bins of bulk erase and Boot Block flash memories are compatible. However, Boot Block devices are functionally different with respect to command writes in two key areas: V_{PP} command write lockout and WE# address latching.

Unlike their bulk erase counterparts, Boot Block flash memories allow command writes regardless of V_{PP} voltage applied to them, as long as V_{CC} is above V_{LKO} and RP# is above V_{IL}. This was done for full access to both the array data, Status Register and device identifiers. Program and erase command sequences written to Boot Block devices with $V_{PP} = V_{PPL}$ will **not** result in data alteration. In such cases, internal automation will immediately terminate with error indication in the Status Register.

Bulk erase flash memories internally latched addresses on the leading (or falling) edge of WE# and data on WE#'s trailing (or rising) edge. Boot Block devices, on the other hand, latch both addresses and data on WE#'s trailing edge, resulting in a much simpler hardware interface.

Specific AC write specification differences between the 28F001BX and 28F010 are shown in Appendix K.

8.0 SUMMARY

This application note has discussed key considerations when converting designs using bulk erase flash memories to Boot Block devices. Specific component information can be found in device datasheets, listed in Appendix L. Contact your local Intel or distribution sales office for more information or to obtain assistance in evaluating bulk erase and Boot Block flash memory alternatives.

LIST OF APPENDICES

A: DIP Pinout Comparison (28F010 and 28F001BX)

B: PLCC Pinout Comparison (28F010 and 28F001BX)

C: TSOP Pinout Comparison (28F010 and 28F001BX)

D: Bulk Erase Flash Memory Program Algorithm

E: Bulk Erase Flash Memory Erase Algorithm

F: Boot Block Flash Memory Automated Program Algorithm

G: Boot Block Flash Memory Automated Erase Algorithm

H: Boot Block Flash Memory Status Register

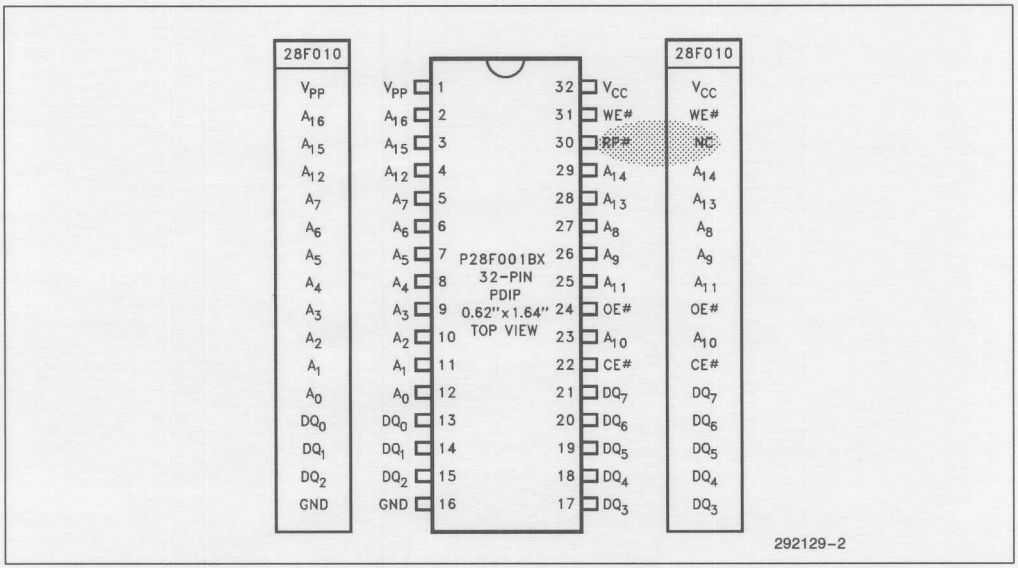
I: DC Specification Comparisons (28F010 and 28F001BX)

J: AC Read Specification Comparisons (28F010 and 28F001BX)

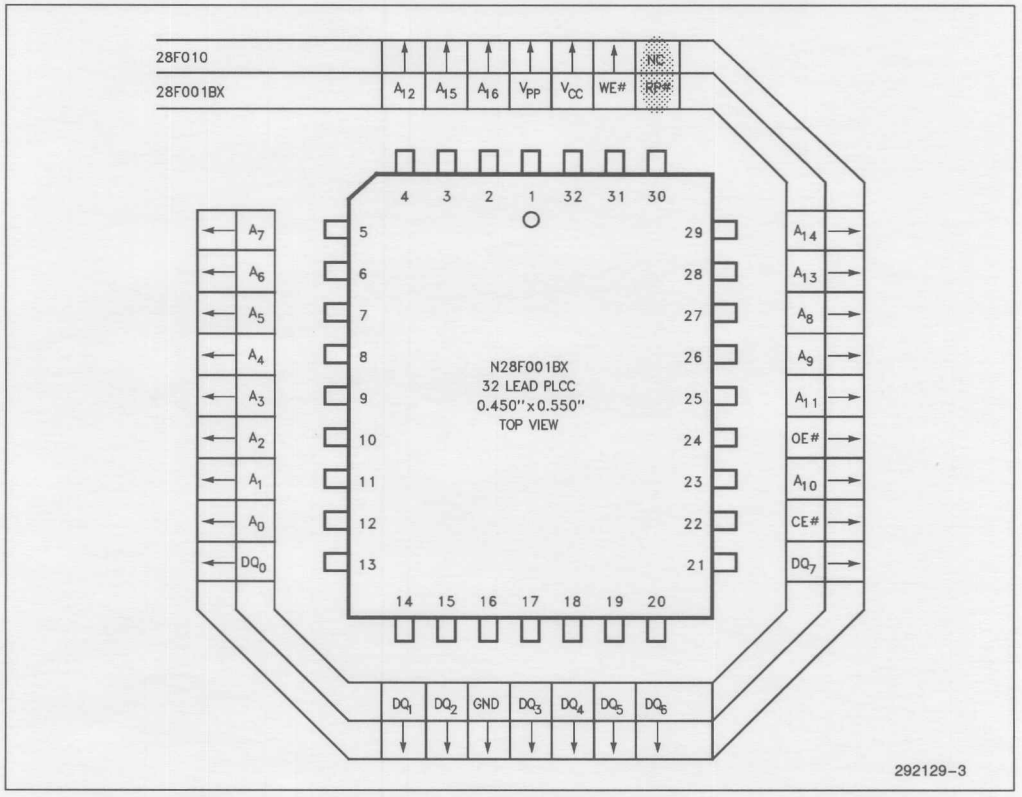
K: AC Write Specification Comparisons (28F010 and 28F001BX)

L: Bulk Erase and Boot Block Flash Memory Datasheets

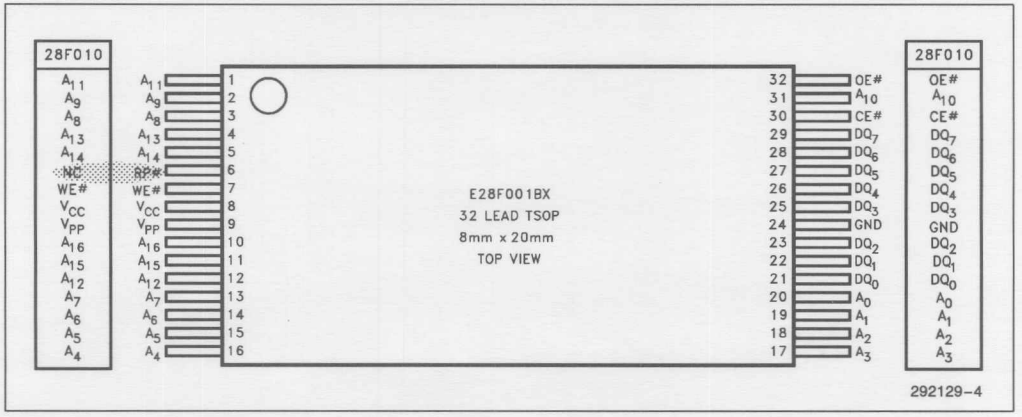
APPENDIX A
DIP PINOUT COMPARISON
(28F010 AND 28F001BX)



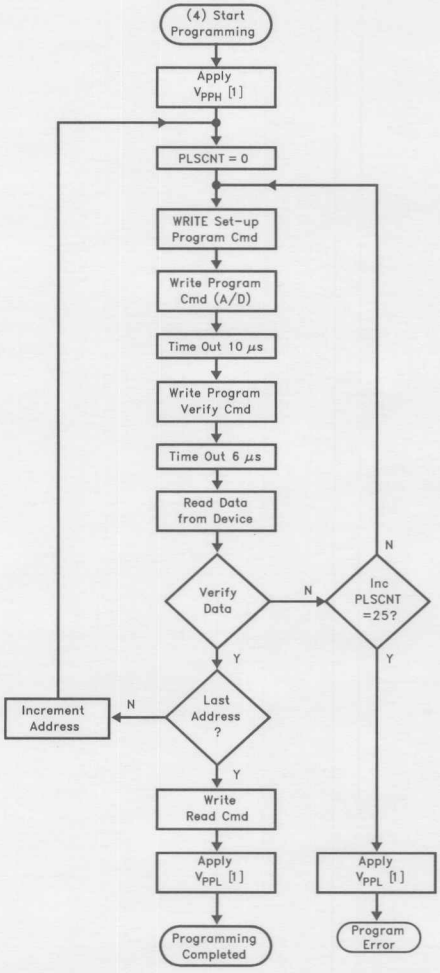
APPENDIX B
PLCC PINOUT COMPARISON
(28F010 AND 28F001BX)



APPENDIX C
TSOP (STANDARD) PINOUT COMPARISON (28F010 AND 28F001BX)



APPENDIX D
BULK ERASE FLASH MEMORY PROGRAM ALGORITHM

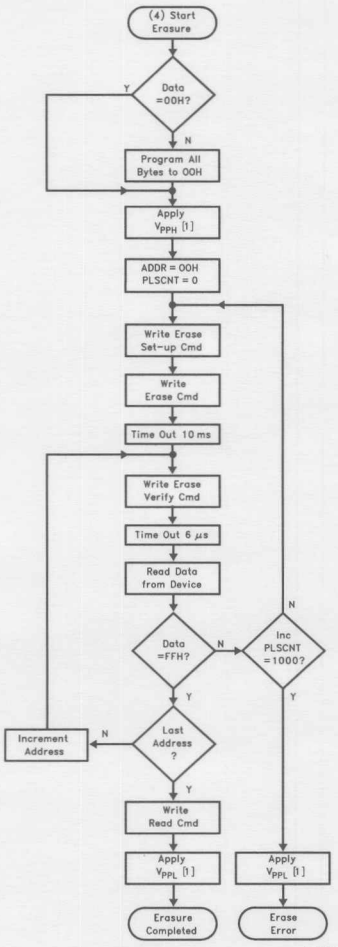


292129-5

Bus Operation	Command	Comments
Standby		Wait for V_{PP} Ramp to V_{PPH}
		Initialize Pulsecount
Write	Setup Program	Data = 40H
Write	Program	Valid Address/Data
Standby		Duration of Programming Operation (t_{WHWH1})
Write	Program Verify	Data = C0H, Stops Program Operation
Standby		t_{WHGL}
Read		Read Byte to Verify Programming
Standby		Compare Data Output to Data Expected
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}

NOTES:
1. See DC Characteristics for the value of V_{PPH} and V_{PPL}.
2. Program Verify is only performed after byte programming. A final read/compare may be performed (optional) after the register is written with the Read command.
3. Refer to principles of operation.
4. **CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.**

APPENDIX E
BULK ERASE FLASH MEMORY ERASE ALGORITHM

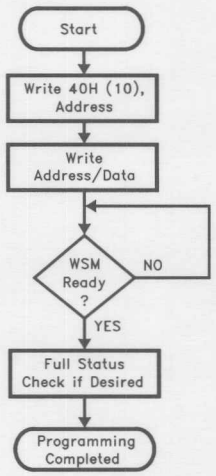


292129-6

Bus Operation	Command	Comments
		Entire Memory Must = 00H before Erasure
		Use Quick Pulse Programming Algorithm
Standby		Wait for V_{PP} Ramp to V_{PPH}
		Initialize Address and Pulse Count
Write	Setup Erase	Data = 20H
Write	Erase	Data = 20H
Standby		Duration of Erase Operation (t_{WHWH2})
Write	Erase Verify	Address = Byte to Verify, Data = A0H, Stops Erase Operation
Standby		t_{WHGL}
Read		Read Byte to Verify Erasure
Standby		Compare Data Output to FFH Increment Pulse Count
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}

NOTES:
1. See DC Characteristics for the value of V_{PPH} and V_{PPL}.
2. Erase Verify is performed only after chip erasure. A final read/compare may be performed (optional) after the register is written with the Read command.
3. Refer to principles of operation.
4. **CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.**

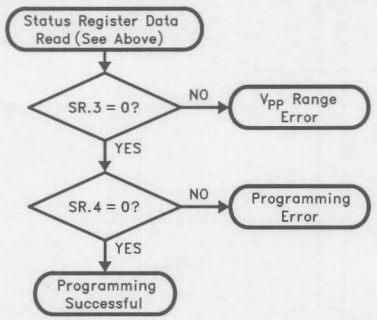
APPENDIX F
BOOT BLOCK FLASH MEMORY AUTOMATED PROGRAM ALGORITHM



292129-7

Bus Operation	Command	Comments
Write	Program Setup	Data = 40H (10H) Address = Location to be programmed
Write	Program	Data to be programmed Address = Location to be programmed
Read		Status Register Data Toggle OE# or CE# to update Status Register Check SR.7 1 = Ready, 0 = Busy
Repeat for subsequent locations Full status check can be done after each location or after a sequence of locations Write FFH after the last program operation to reset the device to Ready/Array Mode		

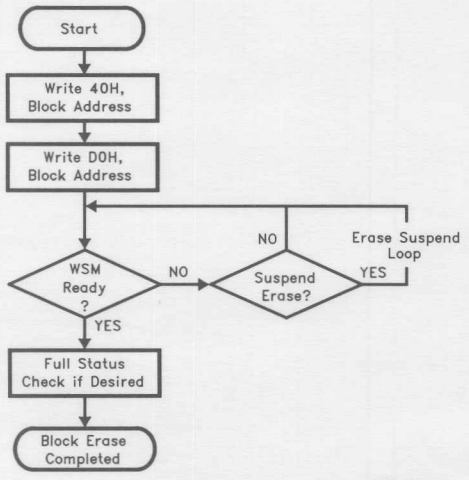
FULL STATUS CHECK PROCEDURE



292129-8

Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4 1 = Program Error
SR.3 MUST be cleared, if set during a program attempt, before further attempts are allowed by the Write State Machine SR.4 is only cleared by the Clear Status Register Command, in cases where multiple locations are programmed before full status is checked If error is detected, clear the Status Register before attempting retry or other error recovery		

APPENDIX G
BOOT BLOCK FLASH MEMORY AUTOMATED ERASE ALGORITHM



292129-9

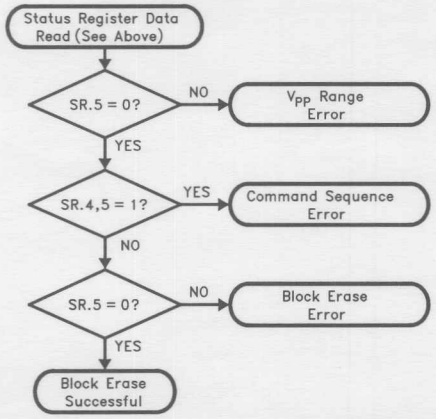
Bus Operation	Command	Comments
Write	Erase Setup	Data = 20H Address = Within block to be erased
Write	Erase	Data = D0H Address = Within block to be erased
Read		Status Register Data Toggle OE# or CE# to update Status Register Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent blocks

Full status check can be done after each block or after a sequence of blocks

Write FFH after the last block erase operation to reset the device to Ready/Array Mode

FULL STATUS CHECK PROCEDURE



292129-10

Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4, SR.5 Both 1 = Command Sequence Error
Standby		Check SR.5 1 = Block Erase Error

SR.3 MUST be cleared, if set during a block erase attempt, before further attempts are allowed by the Write State Machine

SR.5 is only cleared by the Clear Status Register Command, in cases where multiple bytes are erased before full status is checked

If error is detected, clear the Status Register before attempting retry or other error recovery

APPENDIX H
BOOT BLOCK FLASH MEMORY STATUS REGISTER

WSMS	ESS	ES	PS	VPPS	R	R	R
7	6	5	4	3	2	1	0

SR.7 = WRITE STATE MACHINE STATUS
- 1 = Ready
- 0 = Busy

SR.6 = ERASE SUSPEND STATUS
- 1 = Erase Suspended
- 0 = Erase in Progress/Completed

SR.5 = ERASE STATUS
- 1 = Error in Block Erase
- 0 = Successful Block Erase

SR.4 = PROGRAM STATUS
- 1 = Error in Location Program
- 0 = Successful Location Program

SR.3 = V_{PP} STATUS
- 1 = V_{PP} Low Detect; Operation Abort
- 0 = V_{PP} OK

SR.2–SR.0 = RESERVED FOR FUTURE ENHANCEMENTS

These bits are reserved for future use and should be masked out when polling the Status Register.

NOTES:

The Write State Machine Status bit must first be checked to determine program or erase completion, before the Program or Erase Status bits are checked for success.

If the Program AND Erase Status bits are set to "1"s during an erase attempt, an improper command sequence was entered. Attempt the operation again.

If V_{PP} low status is detected, the Status Register must be cleared before another program or erase operation is attempted.

The V_{PP} Status bit, unlike an A/D converter, does not provide continuous indication of V_{PP} level. The WSM interrogates the V_{PP} level only after the program or erase command sequences have been entered and informs the system if V_{PP} has not been switched on. The V_{PP} Status bit is not guaranteed to report accurate feedback between V_{PPL} and V_{PPH}.

APPENDIX I
DC SPECIFICATION COMPARISON (28F010 AND 28F001BX)

DC specification differences (commercial temperature) between the 28F010 and 28F001BX flash memories, or specifications that exist for one device and not another due to functional differences, are shown below.

28F010

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
I_{CCS}	V_{CC} Standby Current (TTL/NMOS)		0.3	1.0	mA	$V_{CC} = V_{CC}$ Max CE# = V_{IH}
I_{CC2}	V_{CC} Program Current		1.0	10	mA	Programming In Progress
I_{CC3}	V_{CC} Erase Current		5.0	15	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current		5.0	15	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current		5.0	15	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
I_{PP4}	V_{PP} Program Verify Current		5.0	15	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current		5.0	15	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress

28F001BX

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
I_{CCS}	V_{CC} Standby Current (TTL/NMOS)		1.2	2.0	mA	$V_{CC} = V_{CC}$ Max CE# = RP# = V_{IH}
I_{CCD}	V_{CC} Deep Powerdown Current		0.05	1.0	µA	RP# = GND ±0.2V
I_{CCP}	V_{CC} Program Current		5	20	mA	Programming In Progress
I_{CCE}	V_{CC} Erase Current		6	20	mA	Erasure in Progress
I_{CCES}	V_{CC} Erase Suspend Current		5	10	mA	Erase Suspended CE# = V_{IH}
I_{PPD}	V_{PP} Deep Powerdown Current		5.0	15	mA	RP# = GND ±0.2V
I_{PPES}	V_{PP} Erase Suspend Current		90	300	µA	$V_{PP} = V_{PPH}$ Erase Suspended
V_{HH}	RP#, OE# Unlock Voltage	11.4		12.6	V	Boot Block Prog/Erase

APPENDIX J
AC READ SPECIFICATION COMPARISON (28F010 AND 28F001BX)

AC read specifications (commercial temperature) for the 28F010 and 28F001BX flash memories are shown below.

28F010

Versions			28F010-120		28F010-150		Unit
Symbol		Characteristic	Min	Max	Min	Max	
t_{AVAV}	t_{RC}	Read Cycle Time	120		150		ns
t_{ELQV}	t_{CE}	Chip Enable Access Time		120		150	ns
t_{AVQV}	t_{ACC}	Address Access Time		120		150	ns
t_{GLQV}	t_{OE}	Output Enable Access Time		50		55	ns
t_{ELQX}	t_{LZ}	Chip Enable to Output in Low Z	0		0		ns
t_{EHQZ}	t_{HZ}	Chip Disable to Output in High Z		55		55	ns
t_{GLQX}	t_{OLZ}	Output Enable to Output in Low Z	0		0		ns
t_{GHQZ}	t_{DF}	Output Disable to Output in High Z		30		30	ns
	t_{OH}	Output Hold from Addresses, CE# or OE# Change, Whichever is First	0		0		ns
t_{WHGL}		Write Recovery Time Before Read	6		6		µs

28F001BX

Versions		$V_{CC} \pm 10\%$	28F001BX-120		28F001BX-150		Unit
Symbol		Characteristic	Min	Max	Min	Max	
t_{AVAV}	t_{RC}	Read Cycle Time	120		150		ns
t_{AVQV}	t_{ACC}	Address to Output Delay		120		150	ns
t_{ELQV}	t_{CE}	CE# to Output Delay		120		150	ns
t_{PHQV}	t_{PWH}	RP# High to Output Delay		600		600	ns
t_{GLQV}	t_{OE}	OE# to Output Delay		50		55	ns
t_{ELQX}	t_{LZ}	CE# to Output Low Z	0		0		ns
t_{EHQZ}	t_{HZ}	CE# High to Output High Z		55		55	ns
t_{GLQX}	t_{OLZ}	OE# to Output Low Z	0		0		ns
t_{GHQZ}	t_{DF}	OE# High to Output High Z		30		30	ns
	t_{OH}	Output Hold from Addresses, CE# or OE# Change, Whichever is First	0		0		ns

APPENDIX K
AC WRITE SPECIFICATION COMPARISON (28F010 AND 28F001BX)

AC write specification (WE#-controlled write, commercial temperature) differences between the 28F010 and 28F001BX flash memories, or specifications that exist for one device and not another due to functional differences, are shown below.

28F010

Versions		$V_{CC} \pm 10\%$	28F010-120		28F010-150		Unit
Symbol		Characteristic	Min	Max	Min	Max	
t_{VPEL}	t_{VPS}	V_{PP} Setup Time to Chip Enable Low	1.0		1.0		µs
t_{ELWL}	t_{CS}	Chip Enable Setup Time before Write	20		20		ns
t_{AVWL}	t_{AS}	Address Setup Time	0		0		ns
t_{WLAX}	t_{AH}	Address Hold Time	50		50		ns
t_{WLWH}	t_{WP}	Write Pulse Width	60		60		ns
t_{WHEH}	t_{CH}	Chip Enable Hold Time	0		0		ns
t_{WHWL}	t_{WPH}	Write Pulse Width High	20		20		ns

28F001BX

Versions		$V_{CC} \pm 10\%$	28F001BX-120		28F001BX-150		Unit
Symbol		Characteristic	Min	Max	Min	Max	
t_{PHWL}	t_{PS}	RP# High Recovery to WE# Going Low	480		480		ns
t_{ELWL}	t_{CS}	CE# Setup to WE# Going Low	10		10		ns
t_{WLWH}	t_{WP}	WE# Pulse Width	50		50		ns
t_{PHHWH}	t_{PHS}	RP# V_{HH} Setup to WE# Going High	100		100		ns
t_{VPWH}	t_{VPS}	V_{PP} Setup to WE# Going High	100		100		ns
t_{AVWH}	t_{AS}	Address Setup to WE# Going High	50		50		ns
t_{WHAX}	t_{AH}	Address Hold from WE# High	10		10		ns
t_{WHEH}	t_{CH}	CE# Hold from WE# High	10		10		ns
t_{WHWL}	t_{WPH}	WE# Pulse Width High	50		50		ns
t_{WHGL}		Write Recovery before Read	0		0		µs
t_{QVVL}	t_{VPH}	V_{PP} Hold from Valid SRD	0		0		ns
t_{QVPH}	t_{PHH}	PWD# V_{HH} Hold from Valid SRD	0		0		ns
t_{PHBR}		Boot-Block Relock Delay		100		100	ns

APPENDIX L
BULK ERASE AND BOOT BLOCK FLASH MEMORY DATASHEETS

Bulk Erase Flash Memories	Order Number
28F256A	290243
28F512	290204
28F010	290208
28F020	290245

Boot Block Flash Memories	Order Number
28F001BX	290406
28F200BX/002BX	290448
28F200BX-L/002BX-L	290449
28F400BX/004BX	290451
28F400BX-L/004BX-L	290450